

CLAIMS

What is claimed is:

- Sub
A2
1. A TCP/IP-based application system, comprising:
an application module that performs predetermined functions based on external requests from an external queue, the external queue being external to the application system and storing the external requests before the requests are fetched into the application system;
a network interaction module coupled to the application module and the external queue (1) to fetch the external requests from the external queue into the application system and (2) to determine which, if any, of the fetched requests will not be processed by the application module based on the processing capacity of the application module and the rate of the external requests arriving at the external queue.
 2. The TCP/IP-based application system of claim 1, wherein the network interaction module rejects those requests determined not to be processed by the application module.
 3. The TCP/IP-based application system of claim 2, wherein the network interaction module rejects the requests not to be processed by closing their connections.
 4. The TCP/IP-based application system of claim 2, wherein the network interaction module rejects the requests not to be processed by returning

a rejection response with a status code.

5. The TCP/IP-based application system of claim 4, wherein the rejection response returned by the network interaction module is a HTTP response.

6. The TCP/IP-based application system of claim 1, wherein the network interaction module also determines which of the fetched requests will be processed first by the application module.

7. The TCP/IP-based application system of claim 1, wherein the network interaction module further comprises:
an internal queue of a predetermined length that receives and stores the external requests fetched from the external queue;

a decision module that (1) causes the external requests to be fetched into the internal queue and (2) determines which of the fetched requests will be processed by the application module and which of the fetched requests will be not processed by the application module based on the processing capacity of the application module and the rate of the external requests arriving at the external queue;

a notification module that rejects the requests that are determined not to be processed.

8. The TCP/IP-based application system of claim 7, wherein the length of the internal queue is at least equal to that of the external queue.

9. The TCP/IP-based application system of claim 7, wherein the notification module rejects the requests that are determined not to be processed by either closing their connections or sending a rejection response with status code.

10. The TCP/IP-based application system of claim 7, wherein the decision module determines which of the fetched requests will not be processed by the application module by

reducing the number of requests to be processed by the application module if new requests are received in the external queue, wherein the number is previously determined and cannot be less than one;

increasing the number of requests to be processed by the application module if no new requests are received in the external queue, wherein the number cannot exceed the length of the external queue;

storing the remaining unprocessed requests in the internal queue;

fetching all new requests from the external queue into the internal queue and causing all of the newly fetched requests that cannot be stored in the internal queue not to be processed.

11. The TCP/IP-based application system of claim 10, wherein the number of requests to be processed is either increased or reduced by a factor of two.

12. The TCP/IP-based application system of claim 1, wherein the

network interaction module determines which of the fetched requests will not be processed by the application module by

reducing the number of requests to be processed by the application module if new requests are received in the external queue, wherein the number is previously determined and cannot be less than one;

increasing the number of requests to be processed by the application module if no new requests are received in the external queue, wherein the number cannot exceed the length of the external queue;

storing the remaining requests in an internal queue;

fetching all new requests from the external queue into the internal queue and causing all of the newly fetched requests that cannot be stored in the internal queue not to be processed.

13. The TCP/IP-based application system of claim 12, wherein the number of requests to be processed is either increased or reduced by a factor of two.

14. In a TCP/IP-based application system, a method of minimizing response time of the application system to external requests, comprising the steps of:

periodically fetching all of external requests stored in an external queue external to the application system into an internal queue of the application system;

determining which, if any, of the fetched requests not to be processed by the application system based on the processing capacity of the application

module and the rate of the external requests arriving at the external queue;
rejecting the requests not to be processed such that the possibility of
dropping a request from the external queue is minimized and the response time
of the application system to the requests is minimized.

15. The method of claim 14, wherein the step of rejecting the requests
not to be processed is performed by closing their connections.

16. The method of claim 14, wherein the step of rejecting the requests
not to be processed is performed by returning a rejection response with a status
code.

17. The method of claim 16, wherein the rejection response is a HTTP
response.

18. The method of claim 14, further comprising the step of determining
which of the fetched requests will be processed first.

19. The method of claim 11, wherein the step of determining which of
the fetched requests are not to be processed further comprises the steps of
reducing the number of requests to be processed if new requests are
received in the external queue, wherein the number is previously determined;
increasing the number of requests to be processed if no new requests are
received in the external queue;
storing the remaining requests in an internal queue of the application

fetching all new requests from the external queue into the internal queue and causing all of the newly fetched requests that cannot be stored in the internal queue not to be processed.

20. The method of claim 19, wherein the number of requests to be processed is either increased or reduced by a factor of two.

Year	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	